SOC-GA 2332 Intro to Stats Lab 8

Wenhao Jiang

10/25/2024

Part 1: Multivariate Regression & Interaction with One Dummy

Dummies

- For categorical variables, we create dummies or convert them to 0 or 1 dummies when we want to include them in a regression model
- Note that for a categorical variable that have n categories, the regression model will only have $n-1$ dummies or categorical variable predictors, because the n^{th} dummy is redundant given that if an observation does not belong to any of the $n-1$ category, then it must be in the n^{th} category
- We call the left-out category the **reference category**
- Question: what if we include all n categories?
- You should always interpret your model coefficients with the reference category in mind. This could get complicated when you have multiple dummy variables, especially when they are interacted in your model

In the case of the dummies representing "race" in the `earnings_df` that we will be using today, we have:

Category	$Dummy_1(black)$	$Dummy_2(other)$
White	0	0
Black	1	0
Other	0	1

Exercise

1. Import `earnings_df.csv` to your environment. Perform the following data cleaning steps: (1) If `age` takes the value 9999, recode it as `NA`; (2) Create a new variable `female` that equals 1 when `sex` takes the value `female`, and equals to 0 otherwise; (3) Create a new variable `black` that equals 1 when `race` is `black` and equals to 0 otherwise; (4) Create a new variable `other` that equals to 1 when `race` is 'other' and 0 otherwise.
2. Use the `describe()` function from the `psych` package to generate a quick descriptive statistics of your data.
3. Now, estimate the following models and display your model results in a single table using `stargazer(m_1, m_2, ..., m_n, type="text")`.

(1) Model 1: earn ~ age (baseline)
(2) Model 2: earn ~ age + edu
(3) Model 3: earn ~ age + edu + female
(4) Model 4: earn ~ age + edu + female + race
(5) Model 5: earn ~ age + edu + female + race + edu*female

4. Write down your prediction equation for Model 5

$$I_i = \beta_0 A_i + \beta_1 E_i + \beta_2 F_i + \beta_3 B_i + \beta_4 O_i + \beta_5 E_i \times F_i$$

5. In Model 5, holding other variables constant, what will be the predicted difference in estimated mean earnings for a white man and a white women?
6. Holding other variables constant, what will be the predicted difference in estimated mean earnings for a white women and a black women?

$$\text{White woman} = \beta_0 \bar{A}_i + \beta_1 \bar{E}_i \text{White woman} = \beta_0 \bar{A}_i + \beta_1 \bar{E}_i + \beta_2 + \beta_5 \bar{E}_i$$

7. Holding other variables constant, what will be the predicted difference in estimated mean earnings for a white man and a black woman?

$$\text{White man} = \beta_0 \bar{A}_i + \beta_1 \bar{E}_i \text{Black woman} = \beta_0 \bar{A}_i + \beta_1 \bar{E}_i + \beta_2 + \beta_3 + \beta_5 \bar{E}_i$$

```
earnings_df <- read.csv("data/earnings_df.csv", stringsAsFactors = F)

## recode age
earnings_df <- earnings_df %>%
  mutate(age = case_when(
    age==9999 ~ NA,
    .default = age
  ))

## recode female
earnings_df <- earnings_df %>%
  mutate(gender = case_when(
    sex == "female" ~ 1,
    .default = 0))

## base R way of doing it
earnings_df$female <- 0
earnings_df[earnings_df$sex=="female", "female"] <- 1

## create black and other
earnings_df <-
  earnings_df %>%
  mutate(black = case_when(
    race == "black" ~ 1,
    .default = 0
  )) %>%
  mutate(other = case_when(
    race == "other" ~ 1,
    .default = 0
  ))
```

```
earnings_df <- earnings_df %>% mutate(male = 1-female)

m1 <- lm(earn ~ age,
         data = earnings_df)

m2 <- lm(earn ~ age + edu,
```

```r
        data = earnings_df)

m3 <- lm(earn ~ age + edu + female,
        data = earnings_df)

m4 <- lm(earn ~ age + edu + female + black + other,
        data = earnings_df)

m5 <- lm(earn ~ age + edu + female + black + other + edu*female,
        data = earnings_df)

stargazer(m1, m2, m3, m4, m5,
          type = "text",
          omit.stat=c("ser", "f","rsq"))
```

```
##
## ================================================================
##                              Dependent variable:
##              ---------------------------------------------------
##                                     earn
##                 (1)       (2)       (3)        (4)        (5)
## ----------------------------------------------------------------
## age          0.134***  0.132***  0.160***   0.158***   0.156***
##               (0.042)   (0.033)   (0.022)    (0.022)    (0.019)
##
## edu                    4.314***  4.500***   4.477***   6.083***
##                         (0.171)   (0.112)    (0.112)    (0.143)
##
## female                          -20.528*** -20.572***   -1.571
##                                   (0.568)    (0.565)    (1.311)
##
## black                                      -2.307***  -2.385***
##                                              (0.623)    (0.557)
##
## other                                        -0.767     -0.946
##                                              (1.137)    (1.017)
##
## edu:female                                            -3.128***
##                                                         (0.199)
##
## Constant     43.888*** 17.786*** 25.439***  26.429***  16.974***
##               (1.917)   (1.817)   (1.207)    (1.230)    (1.254)
##
## ----------------------------------------------------------------
## Observations    980       980       980        980        980
## Adjusted R2    0.009     0.399     0.743      0.746      0.797
## ================================================================
## Note:                                *p<0.1; **p<0.05; ***p<0.01
```

Part 2: Interaction with Two Dummy Variables

Given the following modeling result, please answer the questions.

Table 1:

	Dependent variable:
	earn
college	6.129^{***}
	(0.187)
black	-2.773^{***}
	(0.183)
college:black	1.496^{***}
	(0.340)
Constant	15.077^{***}
	(0.102)
Observations	5,000
R^2	0.290
Adjusted R^2	0.289
Residual Std. Error	5.026 (df = 4996)
F Statistic	679.910^{***} (df = 3; 4996)
Note:	*p<0.1; **p<0.05; ***p<0.01

0. College premium for a White person

$$\text{College White} = 6.129 * 1 + 15.077\text{Non-College White} = 15.077$$

0. College premium for a Black person

$$\text{College Black} = 6.129 - 2.773 + 1.496 + 15.077\text{Non-College Black} = 15.077 - 2.773$$

1. What will be the predicted difference in estimated mean earnings for a white person with a college degree and a black person with a college degree? Whose earnings will be higher?
2. What will be the predicted difference in estimated mean earnings for a white person with a college degree and a black person without a college degree? Whose earnings will be higher?
3. How to interpret the interaction coefficient?
4. How to interpret the intercept?

Plot Predicted Effects

- We can visualize the predicted effects of key predictors using the `predict()` function in base `R`.
- The idea behind this task is to first create a dataframe with values of all the predictors included in the model, with **only the value of your predictor(s) of interest vary within the possible range, whereas other predictors held at their mean.**
- For example, if we want to examine the effect of **education and gender** on earnings, we create a dataframe with a variable `edu` that varies from 0 to 15 with an interval of 1 (so `edu` $= 0, 1, 2, \ldots, 14, 15$), because the possible value of `edu` in our data is integers from 0 to 15 (you can use `summary(your_df)` to check value ranges).

- We repeat this number sequence for another time so that we have **each level of education for both male and female**. So we need to generate `edu` = 0, 1, 2, . . . , 14, 15, 0, 1, 2, . . . , 14, 15. We use `rep(0:15, 2)` to generate this number sequence.
- `rep(x, times)` replicate `x` (a vector or list) for user-defined `times` (in our case, `times = 2`). You can run this in your `R` console to see what number sequence is returned.
- Then, we generate a dummy variable `female` that equals to 0 for male and 1 for female.
- To create a dataframe that have the combination of each level of `edu` and each gender category, we let `female` = 0 for 16 times, and `female` = 1 for 16 times, using `c(rep(0, 16), rep(1, 16))`. You can run this in your R console to see what number sequence is returned.
- For the rest of the predictors, we fix them at their mean. We add `na.rm = T` in the `mean()` function to specify how we want to deal with NA values. If you don't include `na.rm = T`, `mean()` will return NA if your variable contains NAs.

```
## first, we create a dataframe with all predictor variables
## only the key predictor varies, while the others remain at the mean
pred_IV <- data.frame(edu = rep(0:15, 2)) %>%          ## first, create a df with values of your key pre
  mutate(female = c(rep(0, 16), rep(1, 16)),           ## b/c we are looking at interaction effects,
                                                       ## give gender two values, otherwise fix it at me
         age =  mean(earnings_df$age, na.rm = T),    ## fix other variables at mean
         black = mean(earnings_df$black),
         other = mean(earnings_df$other))

## examine the df
head(pred_IV, 5)
```

```
##   edu female      age black other
## 1   0      0 43.26429 0.306 0.068
## 2   1      0 43.26429 0.306 0.068
## 3   2      0 43.26429 0.306 0.068
## 4   3      0 43.26429 0.306 0.068
## 5   4      0 43.26429 0.306 0.068
```

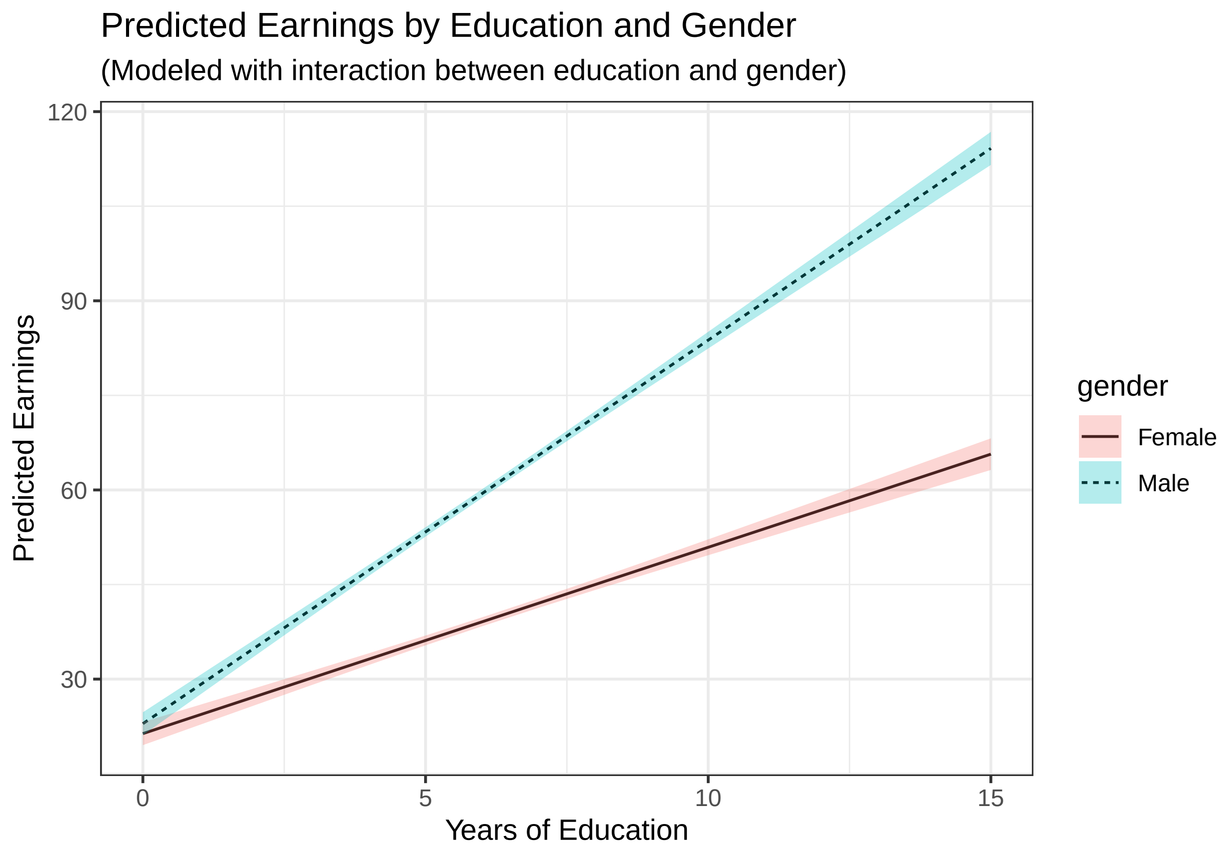
- Now that we have the dataframe `pred_IV` ready for predicting the dependent variable (earning), we can use the R function `predict()` to calculate fitted earning using the regression model and the values specified in each row in `pred_IV`. Then, use `cbind()` to combine this fitted Y value vector with your `pred_IV` for plotting.

```
## use `predict` to predict the Y
predicted_earning <- predict(m5,                       ## the model you are using
                             pred_IV,                  ## the df you use for predicting
                             interval = "confidence", ## set CI
                             level = 0.95)

## bind the columns
pred_result <- cbind(pred_IV, predicted_earning)

## check df
head(pred_result, 5)
```

```
##   edu female      age black other      fit      lwr      upr
## 1   0      0 43.26429 0.306 0.068 22.93127 21.12317 24.73936
## 2   1      0 43.26429 0.306 0.068 29.01392 27.46140 30.56644
## 3   2      0 43.26429 0.306 0.068 35.09657 33.78940 36.40374
## 4   3      0 43.26429 0.306 0.068 41.17922 40.10017 42.25827
```

```
## 5   4       0 43.26429 0.306 0.068 47.26188 46.38025 48.14350
```

```r
## plot
pred_result %>%
  mutate(gender = ifelse(female == 0, "Male", "Female")) %>%       ## convert dummy to character variab
  ggplot(aes(x = edu, y = fit, group = gender)) +
  geom_line(aes(linetype = gender)) +                              ## group linetype by gender
  geom_ribbon(aes(ymin = lwr, ymax = upr, fill = gender), alpha = 0.3) +   # add 95% CI
  theme_bw() +
  labs(x = "Years of Education",
       y = "Predicted Earnings") +
  ggtitle("Predicted Earnings by Education and Gender",
          subtitle = "(Modeled with interaction between education and gender)")
```



Part 3 F-test for Nested Models

- We can use F-test to compare two regression models. The idea behind the F-test for nested models is to check **how much errors are reduced after adding additional predictors**. A relatively large reduction in error yields a large F-test statistic and a small P-value. The P-value for F statistics is the right-tail probability.
- If the F's p-value is significant (smaller than 0.05 for most social science studies), it means that at least one of the additional β_j in the full model is not equal to zero.
- The F test statistic for nested regression models is calculated by:

$$F = \frac{(SSE_{\text{restricted}} - SSE_{\text{full}})/df_1}{SSE_{\text{full}}/df_2}$$

where df_1 is the number of **additional** predictors added in the full model and df_2 is the **residual df for the full model**, which equals ($n - 1 -$ number of IVs in the complete model). The df of the F test statistic is (df_1, df_2).

For example, according to the equation, we can hand-calculate the F value for `m3` vs `m4`:

```
# SSE_restricted:
sse_m3 <- sum(m3$residuals^2)

# SSE_full:
sse_m4 <- sum(m4$residuals^2)

# We add one additional IV, so:
df1 <- 2

# Residual df for the full model (m5):
df2 <- m4$df.residual

# Calculate F:
F_stats <- ((sse_m3 - sse_m4)/df1)/(sse_m4/df2)
F_stats
```

```
## [1] 6.855912
```

```
# Check tail probability using `1 - pf()`
1 - pf(F_stats, df1, df2)
```

```
## [1] 0.001104788
```

- You can also use `anova()` to perform a F-test in R.

```
anova(m4, m5)
```

```
## Analysis of Variance Table
##
## Model 1: earn ~ age + edu + female + black + other
## Model 2: earn ~ age + edu + female + black + other + edu * female
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    974 75711
## 2    973 60403  1     15308 246.59 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- What is your null and alternative hypotheses? What's your decision given the F-test result?

Part 4: Standardized Regression Coefficients

- Why sometimes people report standardized regression coefficients? As we covered in the lecture, the size of a regression coefficient depends on **the scale at which the independent and dependent variables are measured**.
- For example, assume that in a regression model the coefficient of population on the national GDP is 0.0001. This means that 1 additional person will lead to 0.0001 increase in the GDP. However, this value does not necessarily imply that the effect of population is less pronounced than other predictors whose coefficients have a larger value. Because the value of the coefficient depends on the measurement unit of the IV. If we now change population to **population in million**, the new coefficient of population will become $0.0001 \cdot 10^6 = 100$. Although the value of the coefficient gets much larger, this increase is caused by a change in the measurement unit, not the actual effect of population.

- Therefore, it is problematic to use the raw value of the regression coefficient as indicators of relative effect size if the predictors in the model have different measurement units. In such scenarios, standardized regression coefficients can help compare the relative effect size of the predictors even if they are measured in different units.
- Standardized coefficients convert both your dependent variable and independent variables to **z-scores**. That is, each of your original (numeric) variables are converted to have a mean of 0 and a standard deviation of 1. Thus, **standardized coefficients tell us the change in Y, in Y's standard deviation units, for a one-standard-deviation increase in X_i, while holding other Xs constant**.
- There are two methods of getting standardized regression coefficients in R.

Method 1: Use `lm.beta()` from the `QuantPsyc` package

You can get standardized regression coefficients by using the `lm.beta()` function in the `QuantPsyc` package. For example, if we want to get the standardized coefficients for Model 2 (`earn ~ age_recode + edu`):

```
## original model
m2
```

```
##
## Call:
## lm(formula = earn ~ age + edu, data = earnings_df)
##
## Coefficients:
## (Intercept)          age          edu
##     17.7857       0.1321       4.3140
```

```
## standardized coefficients
std_m2 <- lm.beta(m2)
std_m2
```

```
##        age        edu
## 0.09914669 0.62457139
```

- But this method will only report the point estimates instead of a comprehensive modeling result. To obtain that, we need to convert all numeric variables to z-scores and estimate regression models based on the transformed data.

Method 2: Create Z-scores for All Numeric Variables

- For each numeric variables, we create the "standardized variables" by calculating their z-scores:

$$z = \frac{x - \bar{x}}{s_x}$$

- For example, we can use `mutate_at()` to covert numeric variables to z-scores in `earnings_df` using the above formula:

```
## a function that convert a numeric vector to a z-score vector
get_zscore <- function(x){
  (x - mean(x, na.rm = T))/sd(x, na.rm = T)
  }

## create a df with numeric variables converted to z-score
earnings_df_std <- earnings_df %>%
  mutate_at(c("edu", "age", "earn"), get_zscore)
```

```r
## estimate model
m2_std_zscore <-  lm(earn ~ age + edu, data = earnings_df_std)

## compare results
stargazer(m2, m2_std_zscore, type = "text")
```

```
##
## ===========================================================
##                                  Dependent variable:
##                              ----------------------------
##                                         earn
##                                   (1)            (2)
## -----------------------------------------------------------
## age                             0.132***      0.098***
##                                 (0.033)        (0.025)
##
## edu                             4.314***      0.622***
##                                 (0.171)        (0.025)
##
## Constant                       17.786***       0.002
##                                 (1.817)        (0.025)
##
## -----------------------------------------------------------
## Observations                      980            980
## R2                               0.400          0.400
## Adjusted R2                      0.399          0.399
## Residual Std. Error (df = 977)   13.558         0.770
## F Statistic (df = 2; 977)      326.017***    326.017***
## ===========================================================
## Note:                          *p<0.1; **p<0.05; ***p<0.01
```